

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-166941

(43)Date of publication of application : 25.06.1996

(51)Int.Cl.

G06F 17/16
G06F 15/16

(21)Application number : 06-311519

(71)Applicant : HITACHI LTD

(22)Date of filing : 15.12.1994

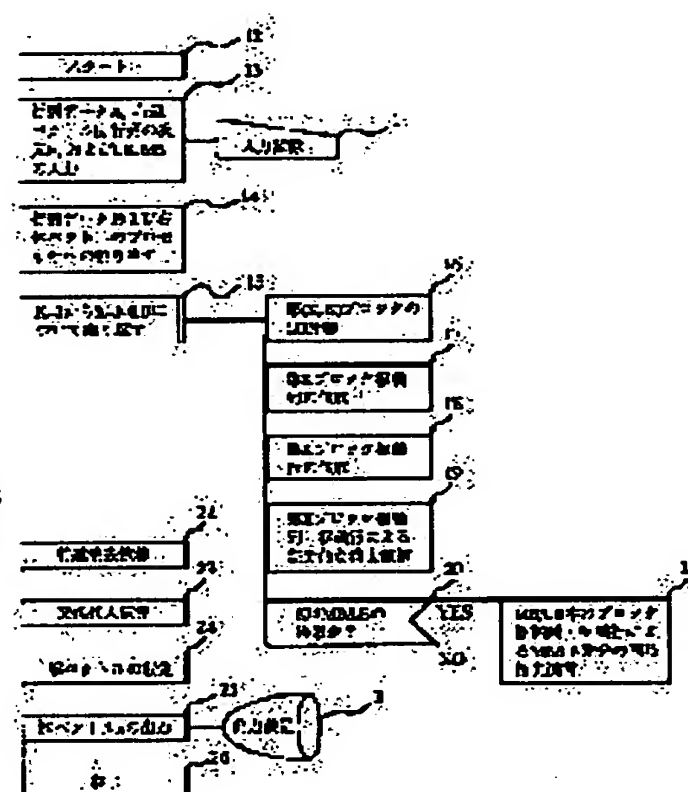
(72)Inventor : YAMAMOTO YUSAKU
OKOCHI TOSHIO
SAGAWA NOBUTOSHI

(54) METHOD FOR LINEAR CALCULATION

(57)Abstract:

PURPOSE: To quicken linear calculation in a parallel computer.

CONSTITUTION: In a system executing linear calculation such as the solving of a linear equation system on the parallel computer, a block size LB determined so that paralleling efficiency becomes maximum and a block size MB determined so that the single performance of matrix multiplication is maximum are inputted (processing 13), a matrix is divided into the blocks of a size LB to assign to a processor (processing 14) and an erasing arithmetic part (processing 15) erases a portion of several block stages collectively (processing 21) so as to process erasing arithmetic as the matrix multiplication of the size MB. Both performance of a single processor and paralleling efficiency are improved like this to quicken linear calculation on the parallel computer.



LEGAL STATUS

[Date of request for examination]

13.12.1999

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

3542184

[Date of registration]

09.04.2004

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

THIS PAGE BLANK (USPTO)

[Date of extinction of right]

2007/03/31

THIS PAGE BLANK (USPTO)

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-166941

(43)公開日 平成8年(1996)6月25日

| | | | | |
|--------------------------|---------|--------|-----------------|--------|
| (51)Int.Cl. ⁶ | 識別記号 | 庁内整理番号 | F I | 技術表示箇所 |
| G 0 6 F 17/16 | | | | |
| 15/16 | 3 9 0 Z | | G 0 6 F 15/ 347 | K |

審査請求 未請求 請求項の数6 O L (全 7 頁)

(21)出願番号 特願平6-311519

(22)出願日 平成6年(1994)12月15日

(71)出願人 000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目6番地

(72)発明者 山本 有作
東京都国分寺市東恋ヶ窪1丁目280番地
株式会社日立製作所中央研究所内

(72)発明者 大河内 俊夫
東京都国分寺市東恋ヶ窪1丁目280番地
株式会社日立製作所中央研究所内

(72)発明者 佐川 暢俊
東京都国分寺市東恋ヶ窪1丁目280番地
株式会社日立製作所中央研究所内

(74)代理人 弁理士 小川 勝男

(54)【発明の名称】 線形計算方法

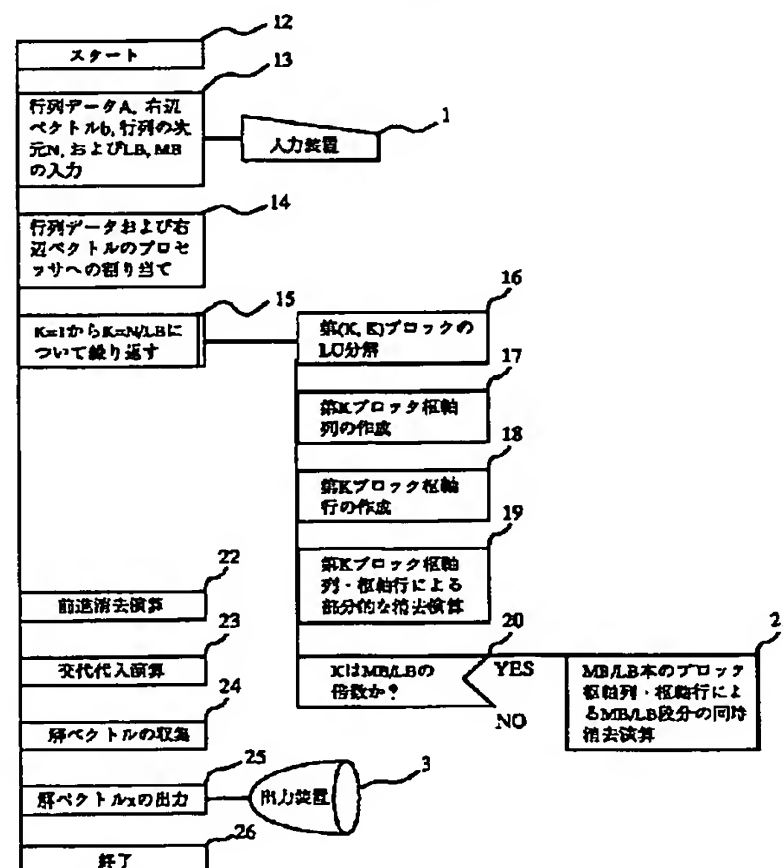
(57)【要約】

【目的】 本発明は並列計算機上での線形計算を高速化することを目的とする。

【構成】 並列計算機上で連立一次方程式の求解などの線形計算を行うシステムにおいて、並列化効率が最大になるように決めたブロックサイズLB、行列乗算の単体性能が最大になるように決めたブロックサイズMBを入力し（処理13）、行列をサイズLBのブロックに分割してプロセッサに割り当て（処理14）、消去演算部分（処理15）では何ブロック段分かの消去をまとめて行うことにより（処理21）、消去演算をサイズMBの行列乗算として処理する。

【効果】 プロセッサ単体での性能と並列化効率とを共に高めることにより、並列計算機上での線形計算を高速化できる。

図5



【特許請求の範囲】

【請求項 1】入力装置と、それぞれがキャッシュメモリを備えた複数のプロセッサからなる処理装置と、出力装置とからなる並列計算機を用いてN次元の行列Aに対して線形計算を行う方法であって、計算対象の行列Aおよび2つの異なるブロックサイズLB、MBを入力し、係数行列Aを大きさLB×LBの複数のブロックに分割して各ブロックを上記プロセッサに割り当て、計算の第K段 ($1 \leq K \leq N/LB$) では、

- 第Kブロック枢軸列および第Kブロック枢軸行を作成するために必要な前処理演算と、
- 第Kブロック枢軸列の作成と、
- 第Kブロック枢軸行の作成と、
- 第Kブロック枢軸列と第Kブロック枢軸行とを用いて更新すべき行列Aの部分のうち、次の数段分のブロック枢軸列とブロック枢軸行を作成するのに必要な範囲の消去演算とを行い、KがMB/LBの倍数であるか否かを判定し、倍数であるときには上記に加えて、
- 行列Aの更新すべき部分の、第 $(K-MB/LB+1)$ 、第 $(K-MB/LB+2)$ 、...、第Kブロック枢軸列・ブロック枢軸行によるMB/LB段分の同時更新演算、を行い、計算終了後に計算結果を出力することを特徴とする線形計算方法。

【請求項 2】請求項 1 の線形計算方法であって、上記入力する処理は、演算に使われるデータを単位にして計った上記キャッシュメモリのサイズをW語とすると、 $MB \times MB \leq W \leq 4 \times MB \times MB$ を満たすように上記MBを決める処理を含む線形計算方法。

【請求項 3】請求項 1 の線形計算方法であって、上記前処理演算は、N次元の連立一次方程式 $Ax = b$ における係数行列Aの第 (K, K) ブロックのLU分解を行う演算である線形計算方法。

【請求項 4】請求項 3 の線形計算方法であって、上記入力する処理は、演算に使われるデータを単位にして計った上記キャッシュメモリのサイズをW語とすると、 $MB \times MB \leq W \leq 4 \times MB \times MB$ を満たすように上記MBを決める処理を含む線形計算方法。

【請求項 5】請求項 1 の線形計算方法であって、上記前処理演算は、N次元の行列Aの第 (K, K) ブロックのQR分解を行う演算である線形計算方法。

【請求項 6】請求項 5 の線形計算方法であって、上記入力する処理は、演算に使われるデータを単位にして計った上記キャッシュメモリのサイズをW語とすると、 $MB \times MB \leq W \leq 4 \times MB \times MB$ を満たすように上記MBを決める処理を含む線形計算方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は並列計算機を用いて例えば連立一次方程式の求解、固有値計算などの線形計算を高速に行う方法に関する。

【0002】

【従来の技術】構造解析、流体計算などの科学技術計算では、数万次元から数百万次元に上る大規模行列を係数行列とする連立一次方程式の求解、固有値計算などの線形計算が必要となる。このような計算を高速に行う手段としては、並列計算機が有力である。並列計算機は数十個から数万個に上る多数の高速プロセッサをネットワークで結んだシステムであり、問題を適切に分割して各プロセッサに割り当て、同時に処理を行わせることにより、1台のプロセッサによる実行と比べて飛躍的な実行速度の向上が可能となる。

【0003】問題の分割と部分問題のプロセッサへの割り当てにおいては、(1) プロセッサの単体性能を発揮させること、(2) 並列化効率 (プロセッサを1台からp台に増やしたときの理想的な性能向上であるp倍に対し、実際にどの程度の性能向上が達成できたかの比) を高めること、の2つを考慮することが必要である。(1)の単体性能に関しては、高性能な並列計算機に使われるプロセッサはキャッシュを備えたRISC型プロセッサであることが多いため、キャッシュを有効に利用できる分割法が必要となる。また、(2)の並列化効率に関しては、プロセッサ間通信の回数とデータ量とを減らして通信時間を削減すること、およびプロセッサ間の負荷を均等にして負荷不均等によるプロセッサの待ち時間を削減することが必要となる。

【0004】連立一次方程式を解くために使われるガウス消去法 (LU分解法) の場合、従来はこの2つの課題を解決するために並列化ブロックガウス法 (たとえばインテル・コーポレーション: "並列処理コンピュータで超大规模連立一次方程式を解く方法", 特開平5-250401 (1991年12月20日出願) 参照) と呼ばれる手法が使われてきた。これは図2の係数行列4を大きさLB×LB (LB: ブロックサイズ) のブロック5、6、7などに分割し、各ブロックを実数のようにみなして、通常の高ス消去法を適用する手法である。並列化にあたっては各ブロックを図2のように16台のプロセッサ0~15に割り当て、各ブロックに対する消去演算を担当のプロセッサに行わせる。

【0005】通常の高ス消去法と比べた場合、並列化ブロックガウス法の利点は次の通りである。まず、消去演算における実数どうしの乗算が、並列化ブロックガウス法ではLB×LBの行列どうしの乗算になる。一般に行列乗算では、データがキャッシュ中にある間に何回も演算を行うため、キャッシュの有効利用が可能となり、RISC型プロセッサ上での単体性能を向上させることが可能となる。さらに、プロセッサ間でのデータ通信もブロック単位となるため、通常の高ス消去法に比べて通信回数は減少し、通信時間の削減により並列化効率の向上が可能となる。これらの利点のため、並列化ブロックガウス法は、並列計算機上でガウス消去法により連立一次方程

式を解くためのもっとも一般的な手法となっている。

【0006】

【発明が解決しようとする課題】並列化ブロックガウス法では、ブロックサイズLBは2つの異なる要請から決定される。まず(1) 単体性能の向上という面からは、消去演算で現われる $C \leftarrow C - A \times B$ という形の行列乗算において、行列A, B, Cのすべてがキャッシュに入るという条件の下で、LBはできるだけ大きいことが望ましい。LBが大きいほど、データがキャッシュ中にある間に多数回の演算が行われることになり、キャッシュの有効利用が計られるからである。一方、(2) 並列化効率の向上という面からもLBの最適値がある。LBが大きくなるとデータ通信の単位となるブロックは大きくなり、通信回数の減少により通信時間が減る一方、各プロセッサの担当するブロックの個数のばらつきが大きくなり、負荷の不均等による待ち時間は増大する。LBが小さくなるとデータ通信回数は増加するが、各プロセッサの負荷は均等になる。したがって、これら2つの均衡により、並列化効率を最大にするLBが定まる。以上で述べた(1) 単体性能を最大にするLBの値と(2) 並列化効率を最大にするLBの値とは、一般には一致しない。例として、16台のプロセッサで並列化ブロックガウス法を実行する場合を図3に示す。この例では単体性能を最大にするようにLBを決めた結果、プロセッサ0の担当ブロック5は4個、プロセッサ15の担当ブロック7は1個と、担当ブロック数に大きな差が生じている。このため、高い並列化効率は期待できない。このように従来の並列化ブロックガウス法では、単体性能、並列化効率の一方あるいは両方を犠牲にしつつ総合的に見てもっとも高い性能を達成するようにLBを決める必要があったため、並列計算機の持つ性能を十分に引き出せないという問題があった。

【0007】本発明は、並列化ブロックガウス法を改良することにより、この問題を解決し、単体性能と並列化効率の両方を高めることにより、線形計算において並列計算機の持つ性能を十分に引き出すことを目的とする。

【0008】

【課題を解決するための手段】上記目的を達成するため、本発明では並列化ブロックガウス法を改良し、2種の異なるブロックサイズを導入した。従来の並列化ブロックガウス法が単体性能と並列化効率とを同時に高められないのは、この2つの目的のそれぞれに対し、最適なブロックサイズが異なるからである。そこで本発明では、まず並列化効率の面から最適なブロックサイズLBを決定し、このサイズにより行列をブロックに分割して、各ブロックをプロセッサに割り当てる。これにより、並列化効率の最適化が可能となる。次に消去演算においては、各ブロック段において毎回行列全体の消去を行うのではなく、ある段数k段分のブロックピボット行とブロックピボット列とを作成しておき、k段に一回、まとめて消去を行う。これにより、消去演算は $MB=k \times LB$ の大き

さの行列乗算として処理できることになり、kを適当に取ることにより、単体性能の最適化が可能となる。

【0009】

【作用】16台のプロセッサでガウス消去法を実行する場合を例にとり、本発明により高い並列化効率と高い単体性能とが同時に達成される様子を説明する。

【0010】まず、図4(a)が全体行列の消去演算を示す。行列はブロックに分割されており、各ブロックのサイズは $LB \times LB$ である。行列中の影の付いたブロックは、プロセッサ0の担当ブロックを示す。LBを十分小さく取ることにより、各プロセッサの担当ブロック数はほぼ均等になり、並列化効率を高めることができる。図4(a)では4段分のブロックピボット列とブロックピボット行とを作成しておき、まとめて4段分の消去を行う場合を示している。この場合、プロセッサ0では、ブロックピボット行・列のうちの影の付いたブロックを用いて消去演算を行う。プロセッサ0の行う演算のみを取り出して表示したのが図4(b)である。図から明らかなように、4ブロック段分の消去をまとめて行うことにより、プロセッサ0での消去演算はサイズ $MB=4 \times LB$ の行列どうしの乗算として処理できる。これにより、従来の並列化ブロックガウス法に比べて行列乗算のサイズを大きくすることができ、キャッシュの有効利用により単体性能を高めることができる。

【0011】以上により、本発明の手法では並列化効率の向上と単体性能の向上とを同時に達成することができる。

【0012】

【実施例】

(実施例1) 以下、本発明の原理および実施例を、図面により詳細に説明する。ここで実施例として挙げるのは、並列計算機を用いてガウス消去法により連立一次方程式を解く方法である。図1に示すように、本方法を適用する並列計算機システムは行列データ、右辺ベクトルなどのデータを入力するための入力装置1、それぞれがキャッシュメモリを備えたp台のプロセッサ31を持つ連立一次方程式求解装置2、解を出力するための出力装置3から構成される。本実施例における処理を図5に示す。まず入力装置から係数行列A、右辺ベクトルb、行列の次元N、2つのブロックサイズLB、MBを入力し(処理13)、行列データおよび右辺ベクトルのプロセッサへの割り当てを行い(処理14)、 $K=1$ から $K=N/LB$ までの各ブロック段について消去演算を行って係数行列のLU分解を求める(処理15)。次に、得られたLU分解を用いて前進消去演算(処理22)、後退代入演算(処理23)を行って解を求め、最後に各プロセッサに分散して格納されている解を収集し(処理24)、出力する(処理25)。本発明のもっとも大きな特徴は、処理14のデータ割り当て部分、および処理15の消去演算部分にあるので、以下、これらの部分について説明する。

【0013】(1) プロセッサへのデータ割り当て
入力ステップ(処理13)で入力したプロセッサ割り当てのためのブロックサイズLBを用いて、図2のように係数行列を大きさLB×LBのブロックに分割し、各ブロックをプロセッサに割り当てる。図2の5、6、7はそれぞれプロセッサ0、1、15に割り当てられたブロックを示す。LBが大きいと通信の単位が大きくなるため、通信回数が減少して通信オーバーヘッドは減少するが、各プロセッサの担当ブロック数のばらつきが大きくなり、プロセッサ間の負荷不均等によりアイドル時間が増大する。一方LBが小さいと負荷は均等になるが、通信の回数が増えて通信オーバーヘッドが増大する。本発明では、これら2つの兼ね合いから、並列化効率を最大にするようにLBをあらかじめ決める。

【0014】(2) 消去演算

本発明における消去演算の特徴は、各ブロック段ごとに行列全体の消去演算を行わず、k本のブロックピボット列・行を予め作成してからkブロック段分の消去を行い、これにより消去演算を大きさMB=k×LBの大きな行列の演算として処理してキャッシュの有効利用を計ることにある。そこで、図5の処理15から処理21により、消去演算の各過程を説明する。

【0015】まず、第Kブロック段の最初では第(K, K)ブロックのLU分解を行う(処理16)。これは従来の並列化ブロックガウス法と同じである。次に、今作成したUの逆行列を第Kブロック列に右からかけることにより第Kブロックピボット列を作成し(処理17)、Lの逆行列を第Kブロック行に左からかけることにより第Kブロックピボット行を作成する(処理18)。そして第Kブロックピボット列と第Kブロックピボット行とにより、消去演算を行う(処理19)。ただし、ここでの消去は行列全体に対しては行わず、次のk本のブロックピボット列・行となる部分に対してのみ行う。k段分のブロックピボット列・行を作成するためには、この範囲の消去で十分である。この後、Kがk=MB/LBの倍数でなければ、第K+1ブロック段に進む(処理20)。Kがk=MB/LBの倍数の場合には、それまでに作成したk本のブロックピボット列・行を用いて、行列全体の消去を行う(処理21)。この消去演算での処理を図4(a)に示す。この例ではk=4であり、4本のブロックピボット列8、4本のブロックピボット行10による消去演算を行っている。図4(a)は全体の係数行列4に対する演算であるが、このうちプロセッサ0の担当ブロック5(それぞれの大きさはLB×LB)に対する演算を抜き出したものを図4(b)に示す。図から明らかなように、ブロックピボット列のうちでプロセッサ0の演算で使われる部分9、ブロックピボット行のうちでプロセッサ0の演算で使われる部分11を用いて、消去演算は大きさMB=4×LBの行列に対する乗算として処理できる。この結果、kを適当に選んでMB×MBの行列3個がちょうどキャッシュに入るようにすれば、キャッシュを最

大限に利用することができ、単体性能が向上する。一方、従来の並列化ブロックガウス法では各ブロック段ごとに行列全体の消去演算を行うので、消去演算はLB×LBの行列乗算となるが、LBの範囲は並列化効率の最適化との兼ね合いにより決められてしまい、単体性能を十分に発揮させることが難しい。本発明では、この点が大きく改良されている。

【0016】(実施例2) 実施例1では線形計算の一つであるガウス消去法の場合について説明したが、これ以外にも、ピボット列とピボット行とを用いて行列を消去していくという線形計算に対しては、本発明が適用可能である。このような線形計算の例としては、最小二乗法などに使われる行列のQR分解や、固有値計算に使われるHouseholder変換などがある。本実施例では、最小二乗法への本発明の適用例を述べる。

【0017】最小二乗法の問題例として、実験により与えられたm個のの入力データと出力データの組(x1, y1), (x2, y2), ..., (xm, ym)に基づき、入力xと出力yの関係をn-1次式 $y = f(x)$ で近似する問題を考える。求めたいのは、f(x)におけるxの0次, 1次, ..., n-1次の項の係数a1, a2, ..., anである。これを求めるには、最小二乗法の標準的な手続きにより、まず第i番目の入力データxiのj-1乗を第(i, j)要素とする行列Aを求め、これを $A = QR$ とQR分解してから、Rを係数行列とし、(Qの転置行列) × (y1, y2, ..., ym)を右辺ベクトルとする連立一次方程式を解けばよい。この一連の処理のうち、もっとも計算量が多いのはAのQR分解の部分であり、これを並列機上で実行する場合に本発明が利用できる。

【0018】本発明の手法を用いて、このQR分解を並列機上で実行する場合の処理を図6に示す。QR分解の場合も、並列化効率を最大にするように選んだブロックサイズLBにしたがって行列をブロックに分割し、プロセッサに割り当てる部分(処理27)は実施例1のガウス消去法と同様である。ただし次の消去演算において、第Kブロックピボット行・列を作成するための前処理演算が、ガウス消去法の場合のLU分解処理(図5の処理16)からQR分解処理(図6の処理28)へと置き替わっている。また、第Kブロックピボット行・列の作成(図6の処理16, 17)の詳細もガウス消去法の場合とは異なる。しかし、あらかじめ数本のブロックピボット行・列を作成しておき、MB/LB段に1回まとめて消去演算を行うという本発明の手法(処理21)は全く同様に適用できる。また、この手法により得られる効果も、ガウス消去法の場合と同様である。

【0019】本実施例では他の線形演算に対する適用例として最小二乗法などに使われるQR分解の場合を示したが、Householder変換など、さらに他の線形計算アルゴリズムに対しても、本手法は同様に適用できる。

【0020】

【発明の効果】以上説明したように、本発明によれば、並列計算機上でのガウス消去法などの線形計算において、単体性能と並列化効率とを共に高めることができ、並列計算機の性能を十分に引き出した計算が可能となる。

【図面の簡単な説明】

【図 1】 並列計算機による線形計算システムの全体を示す図である。

【図 2】 従来の並列化ブロックガウス法でのブロックへのプロセッサ割り当てを示す図である。

【図 3】 従来の並列化ブロックガウス法において、単体性能を高めるためにブロックサイズを大きく取った結果、各プロセッサの担当ブロック数が不均等になったことを示す図である。

【図 4】 本発明により、プロセッサ割り当てのためのブロックサイズを小さくとっても、消去演算のためのブロックサイズが大きく取れるようになったことを示す図である。

【図 5】 本発明をガウス消去法による連立一次方程式解法に適用した場合の手続き全体を示す図である。

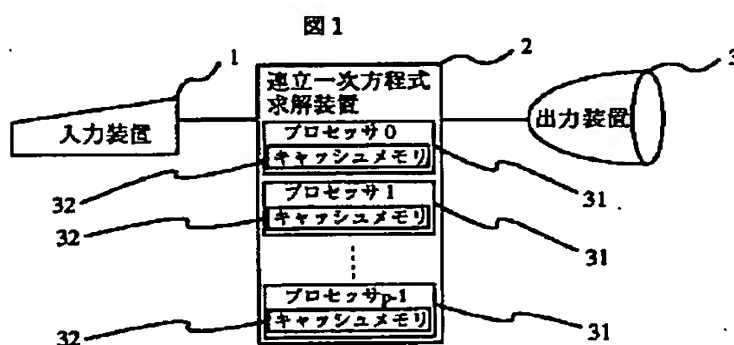
【図 6】 本発明を行列のQR分解に適用した場合の手続き

全体を示す図である。

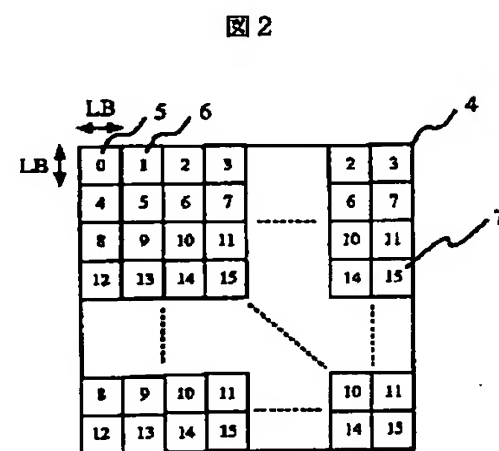
【符号の説明】

1 : 入力装置, 2 : 連立一次方程式求解装置, 3 : 出力装置, 4 : 係数行列の全体, 5 : プロセッサ0の担当するブロック, 6 : プロセッサ1の担当するブロック, 7 : プロセッサ15の担当するブロック, 8 : 4段分のブロックピボット列の全体, 9 : ブロックピボット列のうちプロセッサ0の計算に使われる部分, 10 : 4段分のブロックピボット行の全体, 11 : ブロックピボット行のうちプロセッサ0の計算に使われる部分, 12 : スタート, 13 : データの入力, 14 : 係数行列および右辺ベクトルのプロセッサへの割り当て, 15 : 最外側の繰り返しループ, 16 : 第(K, K)ブロックのLU分解, 17 : 第Kブロックピボット列の作成, 18 : 第Kブロックピボット行の作成, 19 : 第Kブロックピボット列・行による部分的な消去演算, 20 : KがMB/LBの倍数か否かの判定, 21 : K段分の同時消去演算, 22 : 前進消去演算, 23 : 後退代入演算, 24 : 解ベクトルの収集, 25 : 解ベクトルの出力, 26 : 終了, 27 : 行列データのプロセッサへの割り当て, 28 : 第(K, K)ブロックのQR分解, 29 : 解行列の収集, 30 : 解行列の出力, 31 : プロセッサ, 32 : キャッシュメモリ。

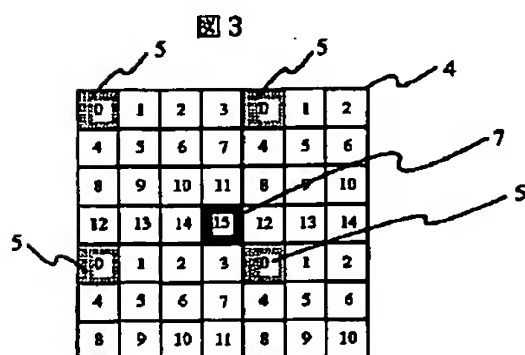
【図 1】



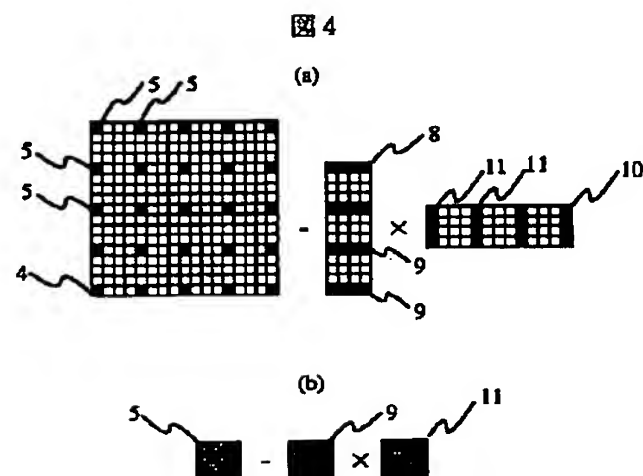
【図 2】



【図 3】

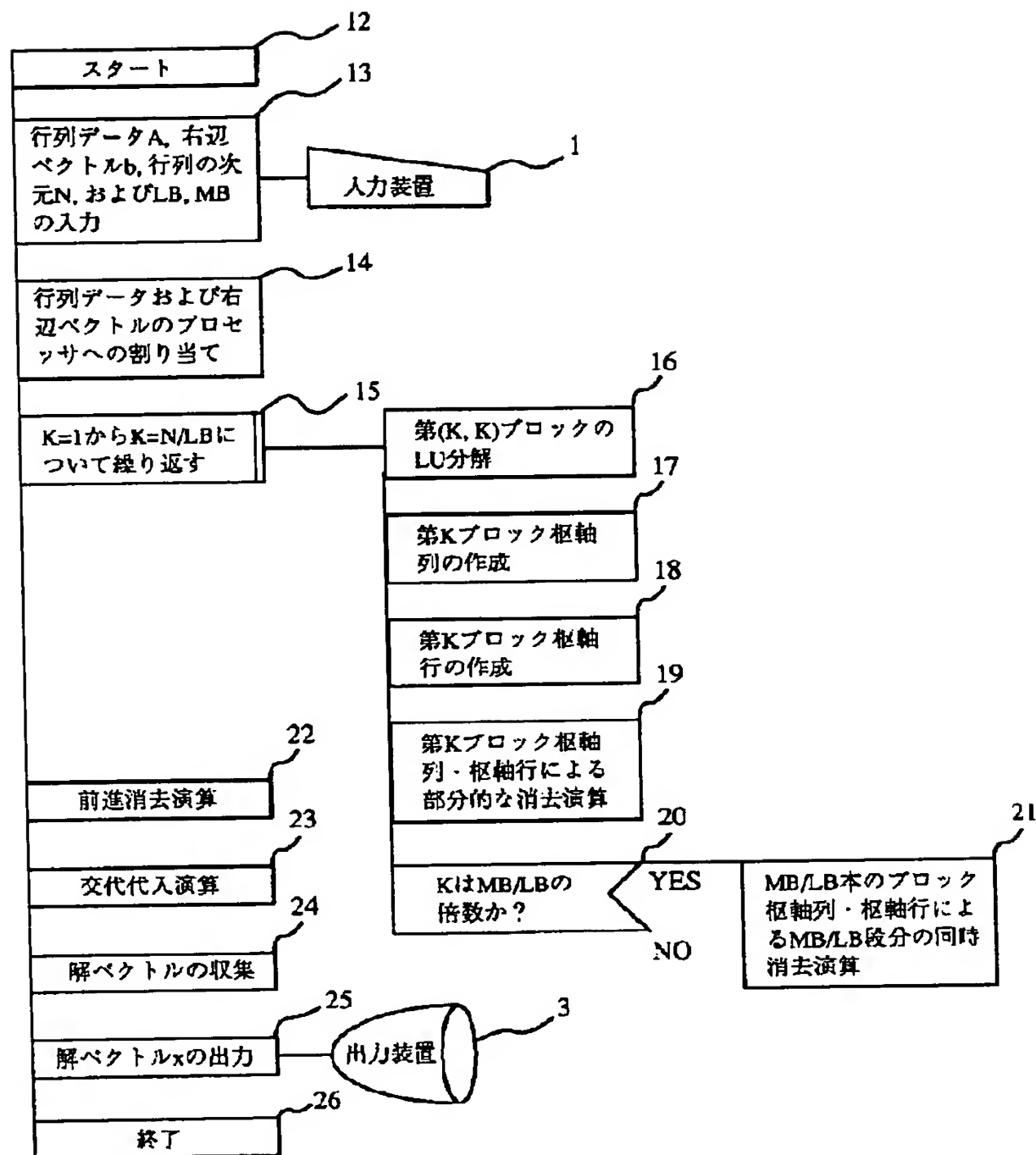


【図 4】



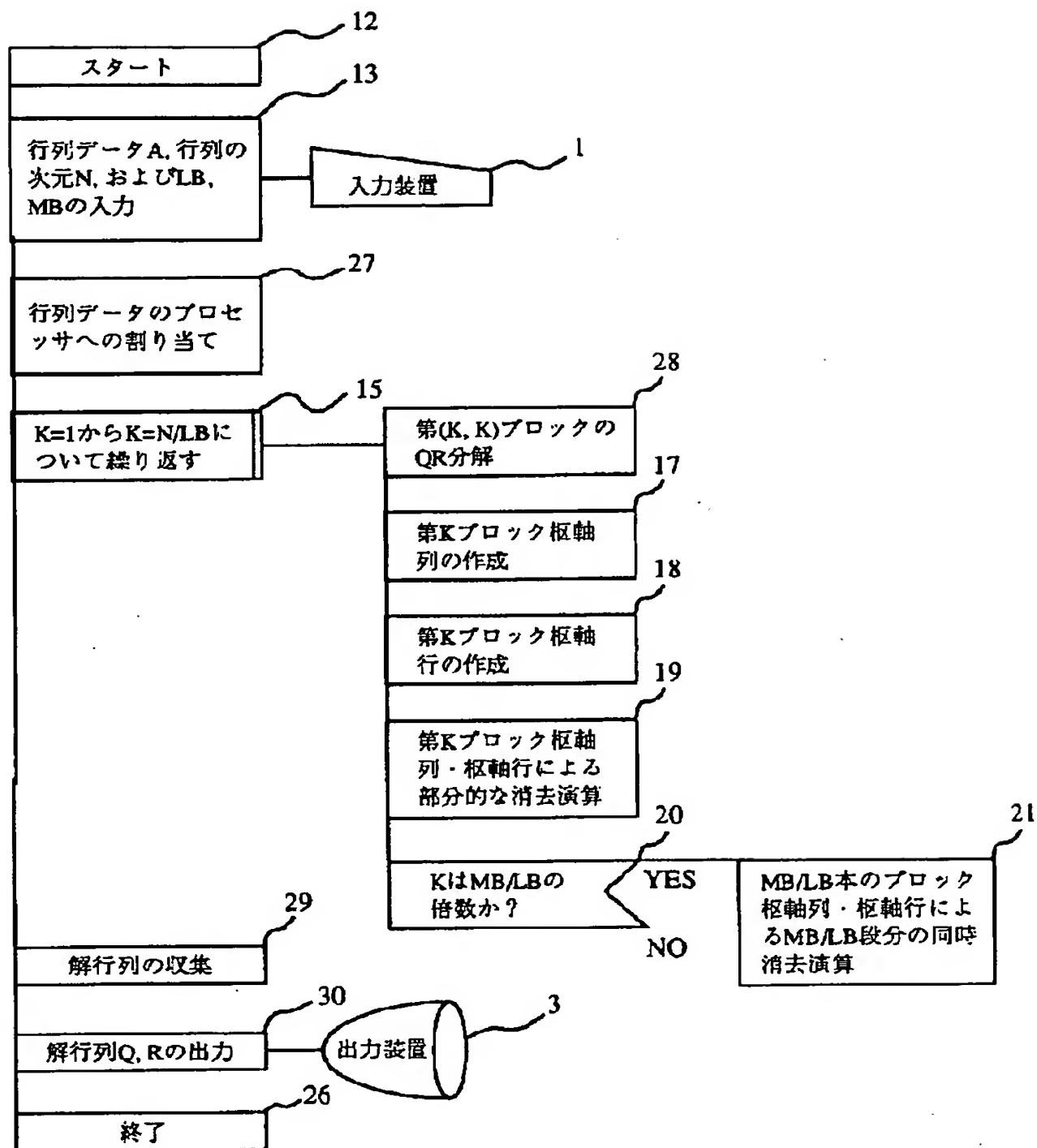
【図 5】

図 5



【図 6】

図 6



THIS PAGE BLANK (USPTO)